NeuraLogix

Data Sheet
NLX230

FUZZY MICROCONTROLLER™

The Fuzzy MicroController (FMC), Model NLX230, is a fully configurable VLSI Fuzzy Logic engine. It is one in a family of MicroController devices offered by NeuraLogix. The FMC devices are intended to augment or supplant conventional microprocessor implementation in performance or cost-critical embedded control systems.

Instead of using algorithms executed sequentially to control an output based on input conditions, as is done in general purpose microprocessor implementations, the devices employ the principles of Fuzzy Logic to calculate an optimal output action based on input conditions, thereby performing a parallel operation to control the output. This efficient implementation allows high processing rates (30 million rules/second) at low cost.

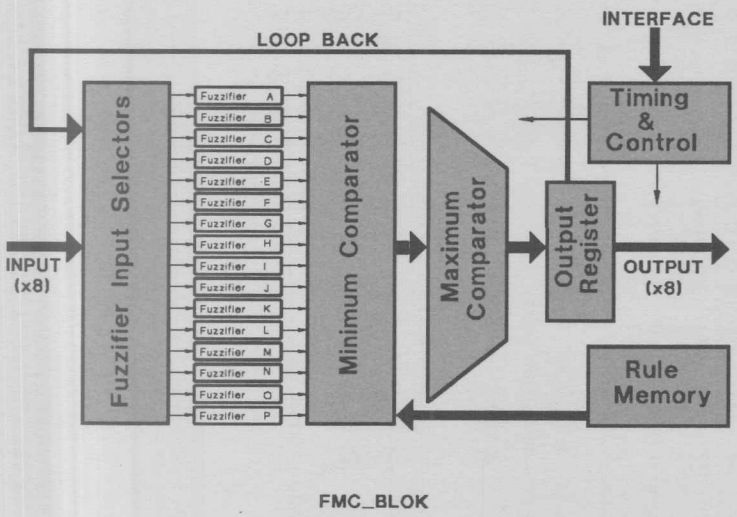
Applications

- Replace Conventional PID Controllers
- Smart Appliances
- Pattern Matching
- Sequencers, State Machines and Timers
- Automotive Applications
- Robotics
- Approximate Reasoning
- Expert Systems

Features

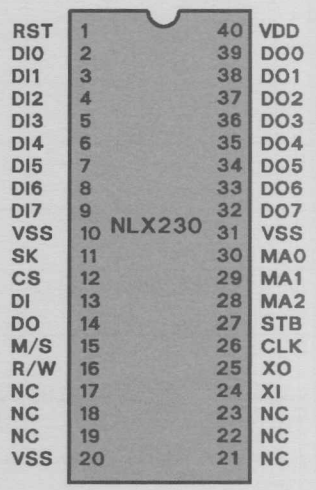
- Powerful Fuzzy Logic Processing (30M Rules/Sec)
- Simple, Low Cost PC Based Development Environment
- Cascadable
- Easy to Configure
- Minimal External Components
- Alternate Packaging Available
- Low Power CMOS

Patent Pending



FMC_BLOK

Fuzzy MicroController



Standard 40-Pin Connection Diagram

CHARACTERISTICS

ABSOLUTE MAXIMUM RATINGS (T_A = +25°C)

Symbol	Parameter	Min	Max	Units	Notes
V_{DD}	DC Supply Voltage	-0.3	7.0	V	
V_{IN}	Input Pin Voltage	0.3	V_{DD}+0.3	V	
I_{IN}	Input Pin Current	-10.0	10.0	µA	@ 25°C
T_{OPR}	Operating Temperature	0	70	°C	
T_{STRG}	Storage Temperature	-55	150	°C	
T_{LEAD}	Lead Temperature	--	300	°C	10 sec

DC CHARACTERISTICS (V_{DD}=5V±5%; T_A=0 to +70°C)

Symbol	Parameter	Min	Max	Units	Notes
V_{IH}	Input High Voltage	2.0	V_{DD}	V	
V_{IL}	Input Low Voltage	-0.3	+0.8	V	
I_{IN}	Input Leakage Current	--	5.0	nA	@ 70°C
I_{IN}	Input Leakage Current Bi-Directional	--	40.0	nA	@ 70°C
I_{IH}	Input High Current	-1.0	--	µA	V_{IN} = 5.5 V_{DD} = 5.5
I_{IH}	Input Low Current	1.0	--	µA	V_{IN} = 0.0 V_{DD} = 5.5
V_{OH}	Output High Voltage	2.4	--	V	V_{DD} = 5.0
V_{OL}	Output Low Voltage	--	0.4	V	V_{DD} = 5.0
I_{OH}	Output High Current	-8.0	--	mA	V_{OH}= 2.4
I_{OL}	Output Low Current	8.0	--	mA	V_{OH}= 0.4
C_{IN}	Input Capacitance	--	5	pF	f=1 MHz
C_{IN}	Input Cap. Bi-Directional	--	15	pF	f=1 MHz
C_{OUT}	Output Capacitance	--	10	pF	f=1 MHz
P_D	Power Dissipation	T.B.D.		W	

PIN DEFINITIONS

PIN	NAME	DESCRIPTION
1	RST	Reset Input, low active
2	DI0	Data Input, bit 0
3	DI1	Data Input, bit 1
4	DI2	Data Input, bit 2
5	DI3	Data Input, bit 3
6	DI4	Data Input, bit 4
7	DI5	Data Input, bit 5
8	DI6	Data Input, bit 6
9	DI7	Data Input, bit 7
11	SK	Serial Clock, Used to clock configuration data into device (Input when M/S = 0, Output when M/S = 1)
12	CS	Chip Select, Used in conjunction with M/S, and R/W to enable Serial data input/output. (Output when M/S = 1, Input When M/S = 0)
13	DI	Serial Data Input
14	DO	Serial Data Output
15	M/Sn	Master/Slave Configuration pin.(Master when M/S = 1, Slave when M/S = 0)
16	R/Wn	Read/Write input pin, used in Slave mode only. (Read = 1, Write = 0)
17 thru 23	NC	These pins MUST be tied to ground.
24	XI	Crystal Oscillator Input
25	XO	Crystal Oscillator Output
26	CLK	Buffered System Clock Output
27	STB	Strobe - Used in conjunction with MA0-MA2 to strobe Input/Output data
28	MA2	Input/Output Mux Address bit 2 output
29	MA1	Input/Output Mux Address bit 1 output
30	MA0	Input/Output Mux Address bit 0 output
32	DO7	Data Output, bit 7
33	DO6	Data Output, bit 6
34	DO5	Data Output, bit 5
35	DO4	Data Output, bit 4
36	DO3	Data Output, bit 3
37	DO2	Data Output, bit 2
38	DO1	Data Output, bit 1
39	DO0	Data Output, bit 0
Power Connections		
10	VSS	Ground
20	VSS	Ground
31	VSS	Ground
40	VDD	+5 volts

DESCRIPTION

FMC LOGIC DESIGN PHILOSOPHY

The FMC was designed around the principles of Fuzzy Logic. Before discussing the details of the operation of the FMC, a brief analysis of the design philosophy is in order.

Fuzzy Logic was initially formulated by L. Zadeh in 1965. Fuzzy Logic is a generalization of multi-valued logic and contains conventional or Boolean Logic as a special case. The theory has been further explicated and developed by many others over the past few years (ref: Kandel, Lee, Sugeno, Yager, etc.).

The primary utility of Fuzzy Logic for control applications is that it lets one construct arbitrarily complex linear or nonlinear functions with intuitively understandable terms and rules. Thus the Fuzzy Logic approach captures the approximate nature of expert human reasoning better than more conventional approaches.

Of course, conventional microprocessor-based approaches also allow the specification of functions with terms and rules. The critical difference with Fuzzy Logic control is that the terms and rules do not evaluate just to the two values, "True" or "False". Rather, fuzzy terms admit to degrees of membership in multiple sets and fuzzy rules may have a continuous range of truth or possibility. In the now classic example, one does not need strictly to be either a member or not of the set "Tall Persons". One may, of course, be "Tall" to a greater or lesser degree, that is a member of the set "Tall Persons" with a degree of membership which lies between 1 and 0. This is basically what is meant by "Fuzziness".

More specifically, consider Figure 1 (after Yamakawa, 1988). This is an artificially simple, but illustrative example of using conventional and Fuzzy Logic techniques to approximate the input/output relationship, Y = f(X). The variable X is the input (possibly from a sensor) and Y is the output (possibly to a motor). There are three ways to specify this relationship:

1) as a mathematical function
2) as a series of points using conventional programming techniques or a lookup table
3) as a set of Fuzzy Logic terms and rules.

In this specific example, the relationship between X and Y could be expressed as Y = 1/(1+e**X). While

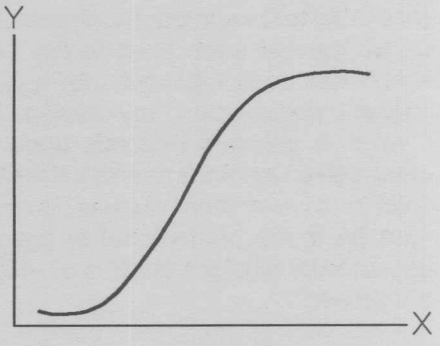


Figure 1.
Conventional and Fuzzy Logic Input/Output Relationship

this is a succinct formulation, it is often impossible to specify real world control problems quite so simply. Note that evaluating this function would require the use of complex mathematics and significant multiplication.

A conventional approach would be to approximate the above relationship between input and output with a ROM-based table or with "if" statements in some programming language. This approach is diagrammed as Figure 2.

While this is certainly feasible, some problems arise. First, a large number of comparisons or table-entries may be required to adequately approximate the

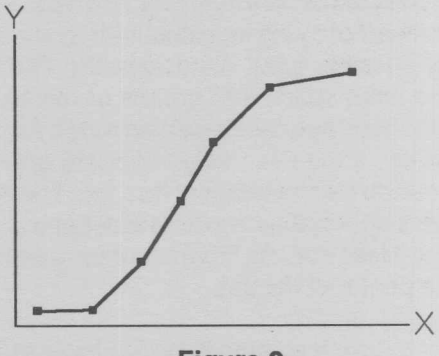


Figure 2.
Conventional Approach to Input/Output Relationship

chosen function. Second, given a limited number of points to approximate the function, how does one deal with values of X which lie between the available points? For a large number of inputs whose values can cover a large range it is impractical to write a rule for every input combination (which is what look up tables do). Thus conventional control systems having a large number of inputs and a large range of values for each input are difficult, if not impractical, to construct.

DESCRIPTION

The essence of the fuzzy approach is diagrammed in Figure 3. Notice that each point in the function approximation has been replaced with a circle of points and that these clusters may overlap. Which output Y value is selected depends upon which cluster center a given X point is nearest. If an X value is equidistant from two or more clusters, its resultant Y value can be made proportional to the output values suggested by all of the clusters of which X is partially a member.

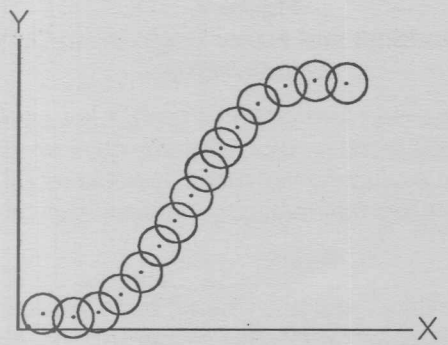


Figure 3.
Fuzzy Logic Approach
Input/Output Relationship

Thus the Fuzzy Logic approach yields a "built in interpolation" which can drastically simplify specification of complex or nonlinear relationships. In other words, instead of trying to account for every combination of possible input conditions the FMC segments the input space into groups of membership functions which represent a certain range (or set) of input values. If the input value falls into one of the sets it is said to be a member of that set. The degree of membership of the input to the set depends on how "far" away it falls from an ideal or center value which lies in the center of the set.

By allowing device outputs to be fed back as inputs (LoopBack), the FMC allows the designer to implement fuzzy sequencers and state machines.

TERMINOLOGY

We term inputs or outputs which yield single, precise values as "crisp". This is in contrast to "fuzzy" rules and membership functions. Since most sensors produce, and most actuators require scalar values, the NLX230 is designed to accept and generate crisp inputs and outputs.

Crisp inputs may be applied to fuzzy "membership functions" to produce "terms". The membership functions determine the size of the clusters in Figure 3. Each membership function specifies a relationship between a crisp input and a fuzzy set. The result of evaluating the conjunction of an input with membership or non-membership of that input value in the elected fuzzy set. This we call a "term" since one may use this resulting value in subsequent fuzzy rules just as one may employ True and Complement in conventional Boolean expressions.

Figure 4 shows two membership functions and a crisp input.

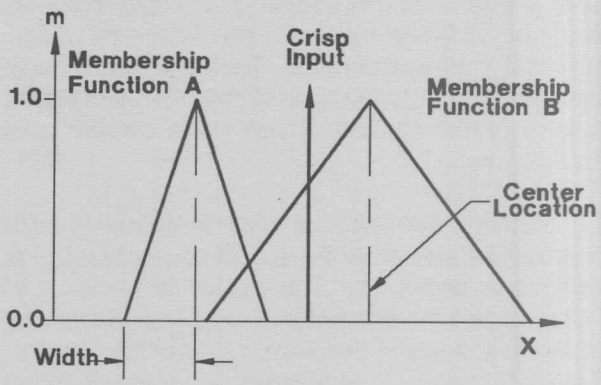


Figure 4.
Membership Functions and Crisp Input

TECHNICAL DESCRIPTION

The FMC is a specialized type of controller. Instead of using algorithms executed sequentially to control an output based on input conditions, the FMC acts in parallel, applying a set of "rules" to a vector of inputs. These rules determine what actions to take based on the input conditions.

Design Technology

The FMC, model NLX230, is a fully configurable VLSI Fuzzy Logic engine. The FMC devices are intended to augment or supplant conventional microprocessor implementations in performance or cost critical embedded control systems.

The devices employ the principles of Fuzzy Logic to calculate an optimal output action based on input

DESCRIPTION

conditions. Input values are ranked by how well they fit within a set of user determined membership functions. For an efficient digital implementation, a linear symmetrical membership function and the simplest min/max fuzzy inference method are utilized.

Rules are used to determine what set of conditions are present at the inputs. Each rule consists of up to sixteen terms, one for each crisp-input/fuzzy membership function pairing, and an Action Value which is the user defined amount by which the output value for that rule is to be modified. The rule which best fits the given set of input conditions determines what modification will be made to the output. The input ranking and rule processing is performed in parallel for all inputs and outputs.

Up to 64 rules can be stored in the on-chip 24-bit wide rule memory. These rules are shared across all of the outputs. Each output may be programmed to use as many rules as needed (up to 64). The number of rules available for any one output depends on the remaining number of rules not being used by the other outputs. An efficient digital implementation of Fuzzy Logic principle allows the NLX230 to achieve high processing rates (30 Million rules/second) at low cost.

Similarity Determination By Distance Measurement

The membership function, MF, in a fuzzy system defines the degree of similarity between an unknown input and a known value. This is traditionally accomplished through the use of an alpha cut (α) and most commonly labeled as mu (μ). Figure 5 demonstrates this principle.

As can be seen, the alpha cut occurs at the point where the fuzzy input [A,8], often referred to as the fuzzy singleton, intersects the membership function. Because the membership function can be of any shape, μ may not always vary in a linear fashion with X.

It has been shown that the actual shape of the membership function is less important than the degree of overlap of membership functions. Determining the optimum membership function shape is not always easy and may only be accomplished empirically in some cases. This is due to the fact that the optimum shape is related to sensor characteristics, control responses, and other dynamic factors.

In order to shield the designer from the membership function shape decision, this design proposes a novel method of degree of similarity determination. Instead of noting the intersection point of the membership function and the fuzzy input singleton, the distance from the center of the membership function to the input is measured. This measurement is performed by subtracting the input from the known Center Location and ignoring the sign. The resulting difference, ac, is then subtracted from the maxium value to obtain d. If the input and center are the same, ac is zero, thus d = MAX, and the further away from the Center Location the smaller is d. Figure 6 indicates that for a membership function with a slope of one, the value d is equal to μ.

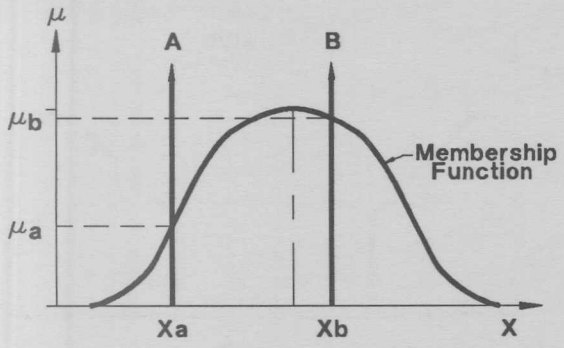


Figure 5. Alpha Cut/Membership Function

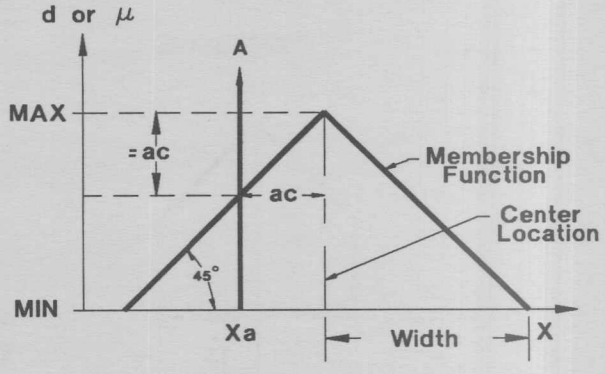


Figure 6. Membership Function Shape

DESCRIPTION

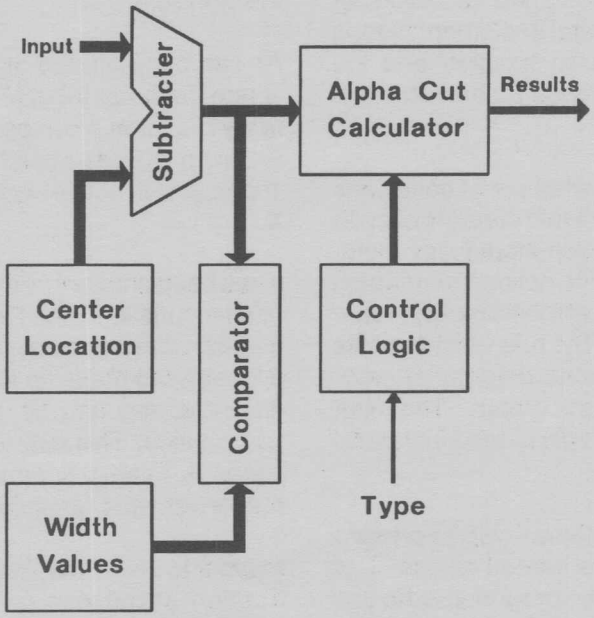


Figure 7. Similarity Circuit

Figure 7 is a block diagram of the circuitry used to determine similarity. The Center Location values and the Input (DI) are passed to the Subtracter, where the difference is calculated. The distance value is compared against the Width values in the Comparator. The Control Logic block is used to control data in the Alpha Cut Calculator in the following way. If the distance is greater than the membership function, the result is forced to zero, or min.

An exception to this is if the Type bit is set, the Type bit causes an exclusive membership function, so if the input is inside, the result would be forced to min.

Figure 8 diagrams Inclusive and Exclusive membership functions with Width set at 13. If data is within the limits set by Center Location and Width, the Alpha Cut Calculator outputs the alpha cut result.

This method is advantageous for several reasons. First, it frees the designer from the need to make the difficult membership function shape decision, while maintaining the most important, degree of overlap information. Secondly, and perhaps most importantly, this method produces an easily implementable solution to the similarity determination problem.

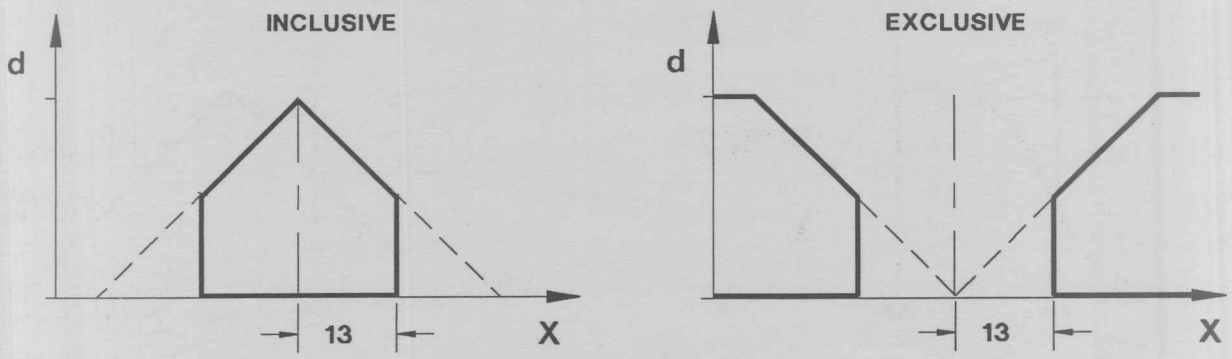


Figure 8. Inclusive and Exclusive Membership Functions

CIRCUIT DESCRIPTION

Block Functions (Reference Figure 9)

Each of the sixteen Fuzzifiers in the NLX230 has a 1-of-8 Selector associated with it. This allows each Fuzzifier to input data from any one of the eight inputs. Each Fuzzifier can look at one and only one input. If a Fuzzifier is configured to use Loop Back from an output, then it may not be used to process an external input. This conjunction of an input with a membership function performed by a Fuzzifier constitutes one "term". The NLX230 supports a maximum of sixteen such terms.

Each Fuzzifier calculates how far the input value is from a user defined "Center Location". The Center Location represents the value the input should be to be considered an exact fit to the membership functions ideal value.

There is one 8-Bit Center Location associated with each Fuzzifier. After calculating the distance, the resulting value is compared to a user defined "Width". The 5-Bit Width value represents the maximum distance the input can be from the Center Location. If the input value's distance from the Center Location is less than or equal to the Width, the input is considered part of the membership function and the distance value is complemented to obtain the degree of membership.

Complementing consists of subtracting the distance from 31 (the maximum width available). The closer the input is to a Center Location, the higher its membership value. Membership values that fall within the Width are passed on to the Minimum Comparator for rule processing.

When the input equals the Center Location value, the output of the Fuzzifier is the maximum (31). When the input value falls outside the selected Width range, the output of the Fuzzifier is zero (0) representing complete lack of membership.

Minimum Comparator

The Minimum Comparator block is implemented with a Neural Network for high-speed throughput. The first sixteen bits of the 24-bit rule are used to enable the outputs of each Fuzzifier for minimum comparisons. Each rule bit has a fixed relationship with a Fuzzifier.

When a rule bit is asserted (1), the corresponding Fuzzifier outputs are compared with one another to find the minimum value.

Maximum Comparator

Once the minimum value for a rule is found, it is stored in a temporary maximum register in the Maximum Comparator. The minimum value for each

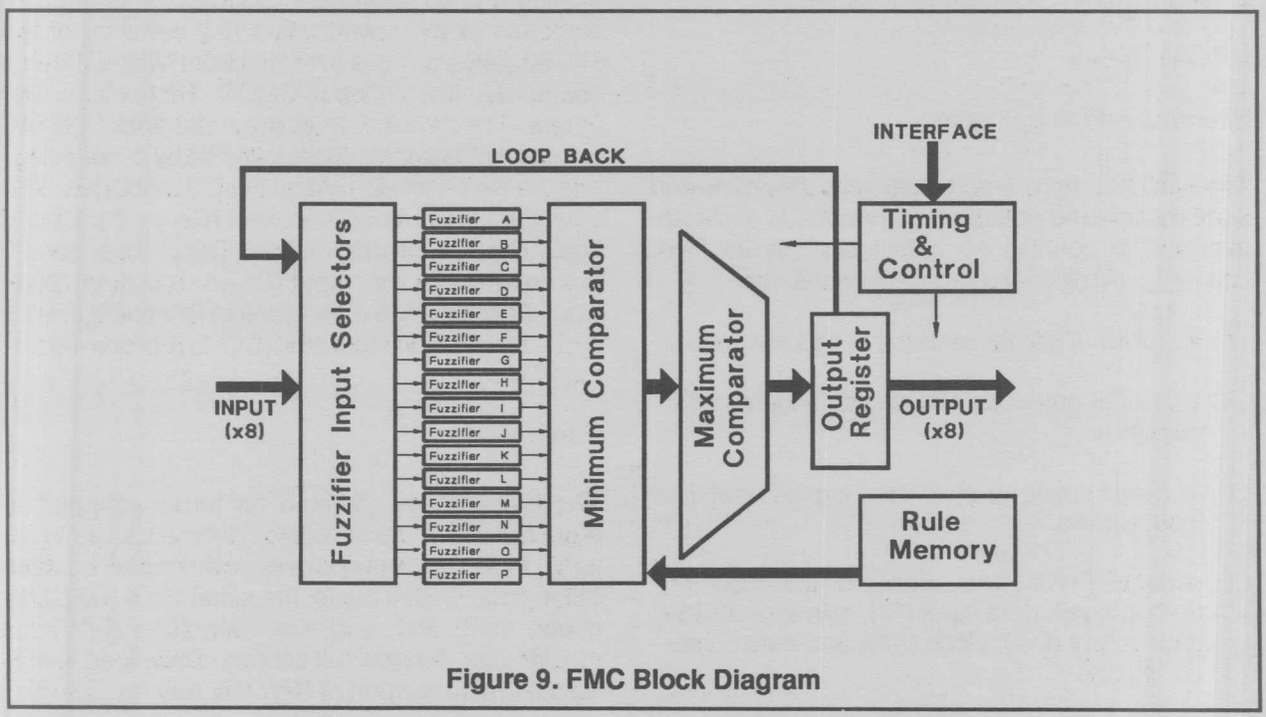


Figure 9. FMC Block Diagram

rule being processed is compared against a reference value stored in the maximum register.

After processing all of the rules, the resulting value in the maximum register is the maximum value of all the minimums. This value represents a "fuzzy" logical sum-of-products. The winning rule's Action Value is passed on to the defuzzification control.

The Action Value is specified in the eight remaining bits of each rule memory word and represents the amount by which the output should be modified. The Action Value is an 8-bit two's complement value ranging from -128 to 127.

Output Register

The 8-bit Action Value from the rule memory word is added to a user-defined Initial Value. For example, an Initial Value of 100 added to an Action Value of -5 would result in an output figure of 95. Saturable ("sticky") arithmetic is used to keep the output value from "rolling over" (modulo 128) in the event adding an Action Value attempts to increment or decrement the Output Value past the upper or lower boundary (-128 and 127).

The Output Registers are pipelined to enhance performance. Once all the rules have been processed and the outputs have been defuzzified, the device begins clocking out the data while simultaneously entering the next group of data inputs.

OPERATION

Interface and Initialization

The FMC has eight time-multiplexed input pins and eight multiplexed output pins. Feedback paths are available to connect an output back to an input internally. Additional pins are provided, for:

- an external crystal oscillator or RC network
- a CLOCK output for clocking data inputs and outputs
- framing signals for synchronizing external data input signals
- serial EEPROM/microprocessor interface consisting of; data input (DI), chip select (CS), data output (DO), clock (SK), and read / write (R/W) pins

Initialization

The initialization is mode dependant. Input Pin M/S provides for two distinctly different configuration modes of operation: the Master Mode and Slave Mode.

In the Master mode, the device automatically downloads the configuration data from an external EEPROM. During this time, the outputs remain at logic 0, until the configuration data is loaded and valid data propagates from the input pins (DI) to the output pins (DO).

In the Slave mode, the device awaits the external control logic to download the configuration data. Following the reset cycle, the device will maintain logic 0 on all outputs (DO) until data propagates from the inputs(DI) to the outputs (DO). *However, the internal configuration data is not valid and the resulting outputs should be considered unknown.* After the device is downloaded by the slave control, a reset to the device will result in valid outputs.

Master Mode

The Master mode (defined as M/S pin strapped to +5 volts) causes the device to begin an automatic download cycle upon the negation of the reset pin (i.e., RST = 1). In this mode the device performs all clock generation (SK), chip enable generation (CS), address generation (DO),and data input, necessary to read a 2048 bit serial EEPROM. This protocol conforms to the specification requirements of the 93C56 series devices available from National Semiconductor, International CMOS Technology and others. The device outputs the serial clock (SK) and asserts a chip enable (CS), followed by a "read flag" and the 7-bit address field on the DO output pin. The device then inputs sixteen data bits on the DI and finally negates the chip enable. The process continues until the FMC has input 128 words of data, (2048 bits). Following the completion of this configuration cycle, Data on the input pins (DI0-7) is processed by the FMC.

Slave Mode

The Slave mode (defined as M/S strapped to ground) causes the device to become a slave to external logic for the purpose of downloading configuration data. In this mode, the serial clock (SK), chip enable (CS) and read write operation (R/W) are provided by the external control. Download is initiated by the assertion of R/W low and the CS input

high prior to the rising edge of the serial clock (SK). Data is then input by the DI input pin on the rising edge of SK. All data bits (2048 bits) are written sequentially followed by the negation of CS. Upload is initiated by the assertion of R/W high and the CS input high prior to the rising edge of the serial clock (SK). Data is then output by the DO output pin on the rising edge of SK. All data bits (2048 bits) are read sequentially followed by the negation of CS. A reset (RST asserted low) may be required to initialize the internal data path after a serial download or upload cycle.

Minimum External Circuitry Required

The FMC is designed for a minimum amount of external support circuitry. In a stand-alone configuration the device needs an external serial EEPROM for storage of the configuration information. To connect the FMC to a conventional microprocessor one only need supply signals on the five pins, DI, CS, DO, SK, R/W, as previously described.

INTERNAL OPERATING MODES

Fuzzifier Modes

There are two ways in which the Fuzzifiers may be used. In typical applications, the outputs of the Fuzzifiers represent the closeness of each input from its Center Location.

Alternatively, the FMC can be configured to use Exclusive Fuzzifiers. By using Exclusive Fuzzifiers, the user can weigh a rule term by how far away from the Center Location the input is.

Input Modes

Input to the FMC can be from the eight time multiplexed input pins or from internal Loop Back from the Outputs.

Output Modes

The FMC provides the capability for each output to be configured as an Immediate Output or an Accumulate Output.

In the Immediate Output mode, the Action Value specified for the winning rule (that rule with the largest minimum term), will be added to the initial value. The original Initial Value is used to calculate the output for every FMC processing cycle.

In the Accumulate Output mode, the new value for an output is the present value summed with the new Action Value (which is determined by the winning rule). In this mode the new output value is retained as the Initial Value for use in the next output calculation. This means that the user-defined Initial value is used only during the calculation of the first (after reset) output value.

EXPANSION

The NLX230 is designed so that multiple devices can be used to expand the number of Inputs, Outputs, Fuzzifiers and Rules.

CLOCK GENERATION

An external clock source may be used to drive the FMC directly by inputting the clock signal to the X! input pin. A clock output pin, CLK, is provided for designs requiring clock distribution. The FMC also contatins an active oscillator circuit for clock generation. Figure 10 show a circuit that will provide oscillator frequencies in the range of 10 to 30 MHz.

REGISTERS AND THEIR FUNCTIONS

All registers can be read or written through the interface.

The register address offset locations start at address location $0C_{HEX}$, bit 96, and end at address location FF_{HEX}, bit 2047. The address locations 00_{HEX} thru $0B_{HEX}$, bits 0 thru 95, are reserved for future use and should be inputted as logic zeros.

The FMC registers are described in the following pages along with a functional description of each.

$$F = \frac{5}{\pi * R1 * C1}$$

Where: F in MHz
R in K Ohms
C in pf
$f_{XTAL} = F/3$

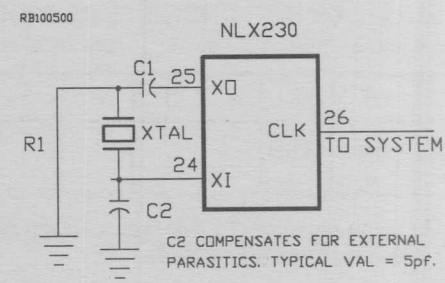


Figure 10. Crystal Oscillator

INTERNAL REGISTERS

OUTPUT MODE REGISTER
Address Offset $0C_{HEX}$ (Bits 96 thru 103)

7	6	5	4	3	2	1	0
Out 7	Out 6	Out 5	Out 4	Out 3	Out 2	Out 1	Out 0

0 = Initial Value + Action
1 = Previous Output + Action

The Output Mode Register determines whether each of the eight outputs will be in the Immediate or Accumulate mode. When an associated bit is set to Logic 0, the output is set equal to the sum of the Initial Value plus the Action Value. When an associated bit is set to Logic 1, the output is set equal to the sum of the Previous Output plus the Action Value.

INPUT CONFIGURATION REGISTER
Address Offset $0D_{HEX}$ (Bits 104 thru 111)

7	6	5	4	3	2	1	0
In 7	In 6	In 5	In 4	In 3	In 2	In 1	In 0

0 = Feedback Output
1 = External Input

The device's input configuration information is contained in this register. Input to a Fuzzifier can be from external Crisp Inputs or from internal Loop Back from the Outputs. When the associated I/O register bit is set to Logic 1, Input is applied data from the Input Pins when the associated I/O Register Bit is set to Logic 0, Inputs are applied from the Loop Back Outputs.

MEMBERSHIP FUNCTION TYPE REGISTER 0 & 1
Address Offset $0E_{HEX}$ (Bits 112 thru 127)

7	6	5	4	3	2	1	0
Fuzzy 7	Fuzzy 6	Fuzzy 5	Fuzzy 4	Fuzzy 3	Fuzzy 2	Fuzzy 1	Fuzzy 0
Fuzzy 15	Fuzzy 14	Fuzzy 13	Fuzzy 12	Fuzzy 11	Fuzzy 10	Fuzzy 9	Fuzzy 8

0 = Inclusive Membership
1 = Exclusive Membership

Each Fuzzifier has associated with it the option of complementing its function. Each register provides control for eight Fuzzifiers. When a bit is set to Logic 0, the function is Inclusive. When a bit is set to Logic 1, the function is Exclusive.

INITIAL VALUE REGISTERS 0 THRU 7
Address Offset 10_{HEX} thru 17_{HEX} (Bits 128 thru 191)

7	6	5	4	3	2	1	0
Sign	MSB						LSB

8-Bit Unsigned initial value for each of the eight outputs.

Each output has an Initial Value register associated with it. Each register contains a user defined Initial Value (range -128 to +127), which corresponds, in the Immediate mode, to the nominal output value for that output with no corrections. In the Accumulate mode the Initial Value is used to calculate the first output after reset.

RULE BOUNDRY REGISTERS 0 THRU 7
Address Offset 18_{Hex} thru $1F_{Hex}$ (Bits 192 thru 255)

7	6	5	4	3	2	1	0
Logic 0	Logic 0	MSB					LSB

6-Bit Rule Number of the last rule associated with like-numbered output.

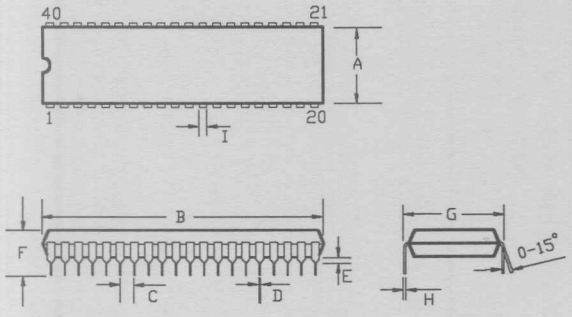
Each output from the device uses a certain number of rules. The number of rules used by each output is determined by the range of contiguous rule addresses which fall between its output rule boundary registers. Each of eight 6-bit boundary registers contains the address of the last rule for its associated output.

Data Sheet

NLX230

DIMENSIONS TABLE

	INCHES		MILLIMETERS	
SYMBOL	MIN	MAX	MIN	MAX
A	0.550	0.550	13.46	13.97
B	-	2.080	-	52.83
C	0.100 BSC		2.54 BSC	
D	0.015	0.025	0.38	0.64
E	0.015	0.060	0.38	1.52
F	0.120	0.175	3.05	4.45
G	0.580	0.620	14.73	15.75
H	0.008	0.015	0.20	0.38
I	0.040	0.060	1.02	1.52



Package Dimensions
40-pin DIP

FMC Development System

A user friendly development system for the FMC devices is available at nominal cost. The PC - based FMC development system allows you to implement your control logic as simple English-like "terms" and "rules". You can then use the development system not only to configure the FMC, but also to test and debug your logic using the available analog and digital I / O provided on the board.

Other Quality Products . . .

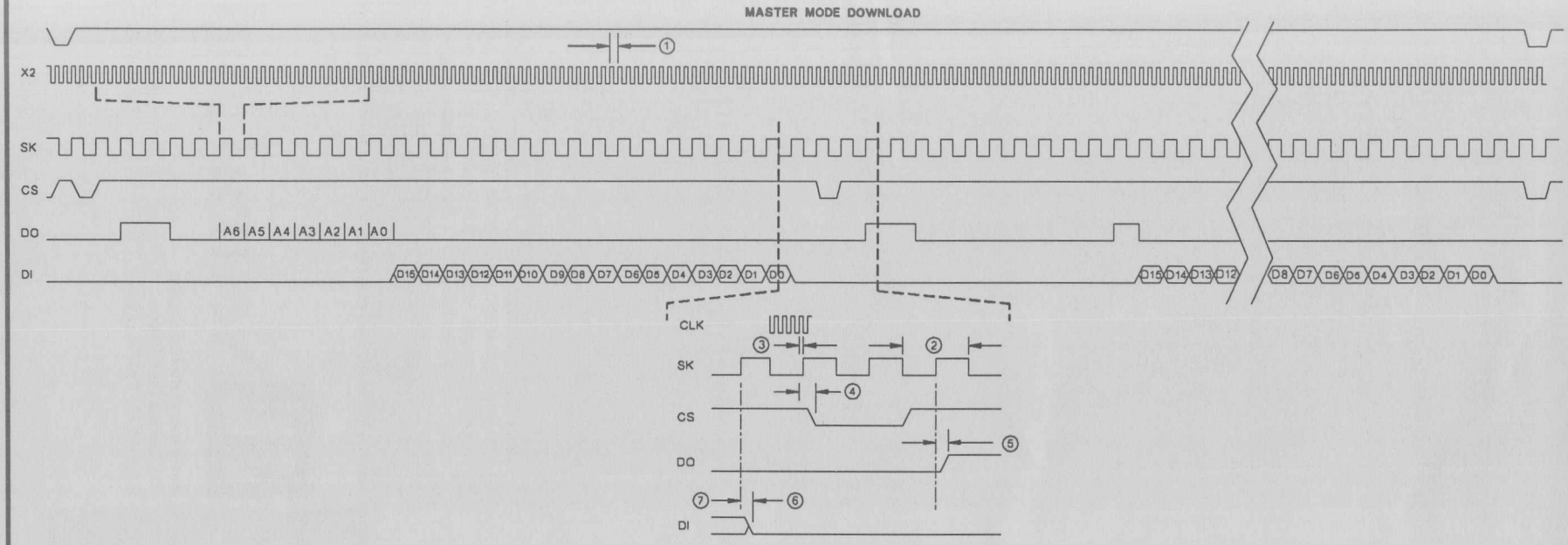
American NeuraLogix, Inc. produces many other fine AI, Neural Network and Fuzzy Logic devices. Call or write for additional information.

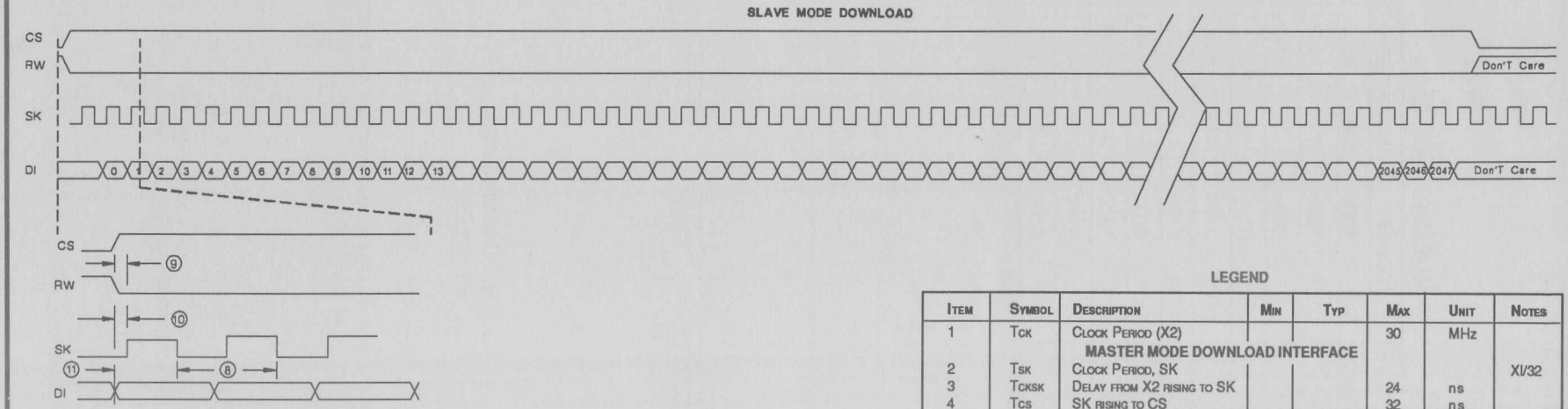
NeuraLogix

American NeuraLogix, Inc
411 Central Park Drive
Sanford, FL 32771 U.S.A.

Tel: 407/322-5608
Fax: 407/322-5609

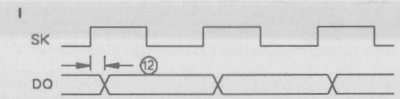
Printed in U.S.A.





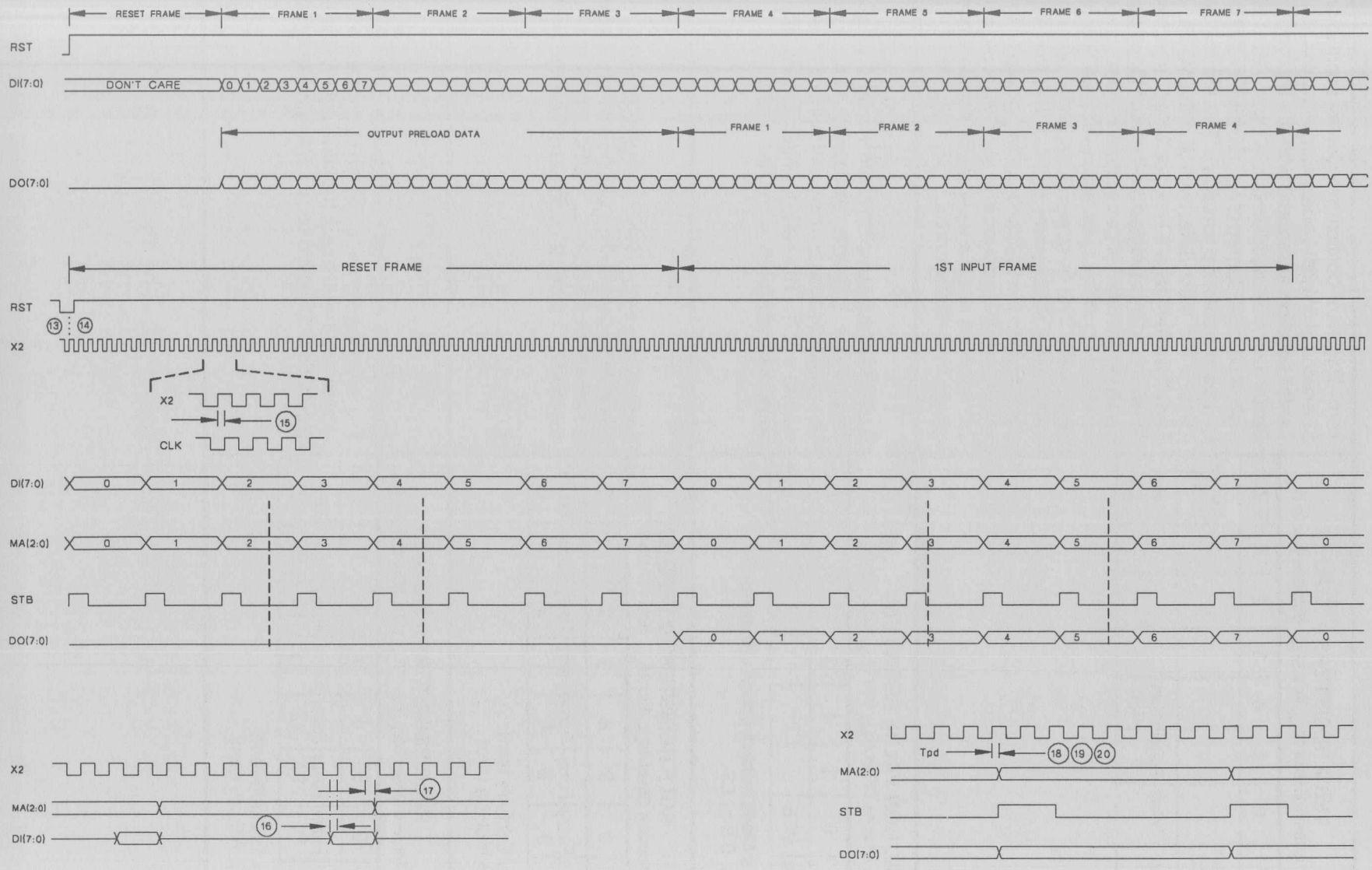
LEGEND

Item	Symbol	Description	Min	Typ	Max	Unit	Notes
1	T_{CK}	Clock Period (X2)			30	MHz	
		MASTER MODE DOWNLOAD INTERFACE					
2	T_{SK}	Clock Period, SK					XI/32
3	T_{CKSK}	Delay from X2 rising to SK			24	ns	
4	T_{CS}	SK rising to CS			32	ns	
5	T_{DO}	SK rising to DO			18	ns	
6	T_{DIS}	DI setup to SK rising	5			ns	
7	T_{DIH}	DI hold	5			ns	
		SLAVE MODE DOWNLOAD/UPLOAD					
8	T_{SK}	Clock Period, Slave Mode (SK)		1000		ns	
9	T_{CSS}	Setup CS to SK	5			ns	
10	T_{RWS}	Setup RW to SK	5			ns	
11	T_{DISU}	Setup DI to SK	5			ns	
12	T_{PDO}	Propagation SK to DO			18	ns	
		DATA INTERFACE					
13	T_{SR}	Setup RST to X2	5			ns	
14	T_{HR}	Hold RST to X2	5			ns	
15	T_{PC}	Propagation X2 to CLK			10	ns	
16	T_{SU}	Setup DI/0-7) to X2	5			ns	



17	T_{HD}	Hold DI(0-7) to X2	5			ns
18	T_{PD}	Propagation X2 to DO(0-7)			38	ns
19	T_{PM}	Propagation X2 to MA(0-2)			24	ns
20	T_{PS}	Propagation X2 to STB			25	ns

SYSTEM TIMING



INTERNAL REGISTERS

TERM REGISTERS 0 THRU 15

Address Offsett 20_{HEX} thru $3F_{HEX}$ (Bits 256 thru 511)

7	6	5	4	3	2	1	0
Center MSB							Center LSB
Select 2	Select 1	Select 0	Width MSB				Width LSB

Term Registers contain the Center Locations, Input Selections and Width values. Each Fuzzifier has a 8-bit Center Location associated with it. The value in the Center Location represents the input value with the highest possible degree of membership for that particular true membership function. The second address contains the Input Selections and the Width values. Each Fuzzifier has a 3-bit Input Select Register which allows the user to program which input (or output Loop Back) the Fuzzifier will use as an input data source. The most significant three bits, binary value (0-7), selects an input (1-8). The least significant 5 bits represent the Width of the membership function; i.e., the maximum distance from the specified Center Location which has any degree of membership at all. Distances outside of this range will result in a degree of membership of zero (0), for Inclusive functions.

ACTION VALUE REGISTERS 0 THRU 63

Address Offset 40_{HEX} thru $7F_{HEX}$ (Bits 512 thru 1023)

7	6	5	4	3	2	1	0
Sign	MSB						LSB

Two's Complement action values for rules 0 thru 63

Each Rule has associated with it a twos complement Action Value. The most significant bit contains the sign. The following seven bits contain the Action Value (range -128 to +127). The Action Values pertain only to the output with which this rule is associated.

RULE REGISTERS 0 THRU 63

Address Offset 80_{HEX} thru BF_{HEX} (Bits 1024 thru 1535)

7	6	5	4	3	2	1	0
Fuzzy 7	Fuzzy 6	Fuzzy 5	Fuzzy 4	Fuzzy 3	Fuzzy 2	Fuzzy 1	Fuzzy 0

1 = Exclude term
0 = Include term

Each Rule is made up of a selection of the Fuzzifiers. The Rule Register selects which Fuzzifiers are included or excluded in the Rule. If the bit is Logic 0, the Fuzzifier is excluded. If the bit is Logic 1, the Fuzzifier is included.

RULE REGISTERS 0 THRU 63

Address Offset $C0_{HEX}$ thru FF_{HEX} (Bits 1536 thru 2047)

7	6	5	4	3	2	1	0
Fuzzy 15	Fuzzy 14	Fuzzy 13	Fuzzy 12	Fuzzy 11	Fuzzy 10	Fuzzy 9	Fuzzy 8

1 = Exclude term
0 = Include term

Each Rule is made up of a selection of the Fuzzifiers. The Rule Register selects which Fuzzifiers are included or excluded in the Rule. If the bit is Logic 0, the Fuzzifier is excluded. If the bit is Logic 1, the Fuzzifier is included.